

SAIMAAN AMMATTIKORKEAKOULU
Tekniikka Lappeenranta
Tietotekniikan koulutusohjelma
Ohjelmistotekniikka

Jonne Frantsi

VARJOSTINLIIKKEEN WWW-SIVUSTO

Opinnäytetyö 2010

TIIVISTELMÄ

Jonne Frantsi

Varjostinliikkeen www-sivusto, 41 sivua

Saimaan ammattikorkeakoulu, Lappeenranta

Tekniikka, Tietotekniikan koulutusohjelma

Ohjelmistotekniikka

Ohjaaja: tuntiopettaja Jouni Könönen

Opinnäytetyön tarkoitus on esitellä käyttökelpoisia tekniikoita dynaamisen verkkosovelluksen toteuttamiseen. Työssä käydään läpi varjostinliikkeen verkkosivuston toteutuksessa käytetyt tekniikat sekä sivuston suunnittelutyö. Sivuston sisällöstä oli tarkoitus tehdä selaimen avulla muokattava. Sivusto on toteutettu ASP.NET-tekniikalla ja C#-ohjelmointikielellä MVC-arkkitehtuuria käyttäen. Merkintäkielenä käytettiin XHTML:ää. Tietokantajärjestelmänä toimii Microsoft SQL Server 2008, jonka kanssa hyödynnetään LINQ to SQL -komponenttia.

Suunnittelua varten tutustuttiin uusiin tekniikoihin. Tiedon lähteinä käytettiin kahden eri kirjaa ja lukuisia Internet-lähteitä. Sivuston toteutukseen kuului kaksi eri osa-aluetta: hallintaosio ja julkinen osio. Suunnittelutyö aloitettiin julkisivun ulkoasusta ja sivupohjan tekemisestä. Tämän jälkeen suunniteltiin tietokanta ja toteutettiin hallintaosio. Viimeisenä tehtiin loppuun julkisivu.

Lopputuloksena saatiin kaikin puolin toimiva verkkosivusto. Julkisivun ulkoasu toimii eri selaimilla moitteettomasti. Hallintaosio on toimiva ja täyttää vaatimukset, jotka sille asetettiin. Hallintaosion avulla voidaan hallita sivuston sisältöä ja varjostinmallien tietoja sekä lukea vierailijoiden palautteita. Hallintaosion käyttö vaatii käyttäjätunnuksen ja salasanan.

Asiasanat:

XHTML, ASP.NET, C#, MVC, Microsoft SQL Server, LINQ to SQL, dynaaminen verkkosovellus

ABSTRACT

Jonne Frantsi

Web Site for Lampshade Manufacturing Company, 41 pages

Saimaa University of Applied Sciences, Lappeenranta

Technology, Degree Programme in Information Technology

Software Engineering

Bachelor's Thesis 2010

Instructor: Mr. Jouni Könönen, Saimaa University of Applied Sciences

The purpose of this thesis was to demonstrate useful technologies for developing a dynamic web application. This thesis covers technologies which were used for the planning and developing the company's web site. The site content was supposed to be editable with a browser. Used technologies were ASP.NET, C# and MVC architecture. The used markup language was XHTML. Microsoft SQL Server 2008 was used as a relational model database server. Database queries were made with a LINQ to SQL -component.

New technologies were studied for the planning stage. Two different books and several Internet sites were used as information sources. The site consists of two different sections: a control panel and a public section. Planning started from making of the public section's graphics and layout. After that the database was designed and the control panel was developed, lastly the public section was finished.

The result of this thesis was a fully functional website that meets the original requirements. The control panel can be used for editing the site content and information about lampshades, and also for reading visitors' feedbacks. The control panel login requires a specific user name and a password.

Keywords:

XHTML, ASP.NET, C#, MVC, Microsoft SQL Server, LINQ to SQL, Dynamic Web Application

SISÄLTÖ

1	JOHDANTO	7
2	TEKNIIKAT	8
2.1	ASP.NET-verkkosovellus	8
2.2	XML	9
2.3	XHTML	11
2.4	CSS	12
2.5	Visual Studio	14
2.6	SQL	14
2.7	LINQ to SQL	17
2.8	jQuery	19
3	SUUNNITTELU	20
3.1	Sivuston ulkoasu	20
3.2	Tietokantarakenne	20
3.3	Sivuston hallintaosio	22
3.4	Sivuston julkinen osio	32
3.5	Julkaisu	35
4	POHDINTA	37
	KUVAT	39
	LÄHTEET	40

TERMIT

.NET Framework	Microsoftin kehittämä ohjelmointiarkkitehtuuri työasema- ja verkkosovellusten kehittämiseen.
Ajax	Asynchronous JavaScript And XML. Joukko verkkosovelluskehityksen tekniikoita, joilla verkkosovelluksista saadaan vuorovaikutteisempia.
ASP.NET	Active Server Pages .NET. Microsoftin kehittämä palvelinperustainen verkkosivujen ohjelmointitekniikka.
CSS	Cascading Style Sheets. Tyyliohje, jolla vaikutetaan HTML-sivun ulkoasuun.
DOM	Document Object Model. Ohjelmointirajapinta, jonka avulla voidaan muokata (X)HTML-dokumentin sisältöä.
Dynaaminen verkkosivu	Verkkosivu, joka luodaan selaimen pyynnöstä.
Dynaaminen verkkosovellus	Verkkosovellus, joka käsittelee selaimen pyyntöjä ja luo dynaamisia verkkosivuja.
FTP	File Transfer Protocol. Protokolla, jonka avulla siirretään tiedostoja Internetin yli.
HTML	HyperText Markup Language. Verkkosivujen tekemisessä käytetty merkintäkieli.
JavaScript	Olioperustainen kieli, jolla voidaan ohjelmallisesti käsitellä verkkosivun objekteja.
LINQ	Language Integrated Query. Komponentti, joka mahdollistaa datakyselyt .NET-ohjelmointikielessä.
MS SQL Server	Microsoft SQL Server. Microsoftin kehittämä relaatiotietokannan hallintajärjestelmä.
MVC	Model-View-Controller. Ohjelmistoarkkitehtuuri, jolla erotetaan käyttöliittymä sovelluslogiikasta.
PHP	PHP: Hypertext Preprocessor. Palvelin pohjainen ohjelmointikieli, jota käytetään mm. dynaamisten verkkosivujen

SQL	jen luonnissa. Structured Query Language. Standardoitu kyselykieli tietokannassa olevan tiedon käsittelyyn.
Web Forms	Ohjelmointimalli, joka käsittää verkkosovelluksen käyttöliittymäkomponentit.
Visual C# / VB.NET / J#	Microsoftin kehittämä ohjelmointikieli, osa Microsoftin .NET-konseptia.
XHTML	eXtensible Hypertext Markup Language. XML:ään pohjautuva verkkosivujen tekemisessä käytetty merkintäkieli.
XML	Merkkauskieli, jolla voidaan kuvata tiedon merkitys tiedon sekaan.

1 JOHDANTO

Tässä opinnäytetyössä käydään läpi verkkosovelluksen toteutusta ja siihen liittyviä tekniikoita. Työ tehdään, koska yrityksellä ei ole verkkosivustoa. Työ ei keskity sovelluksen tekniseen toteutukseen, vaan pääosin sivuston käyttöön ja toimintaan. Työssä esitellään käyttökelpoisia tekniikoita dynaamisen verkkosovelluksen toteuttamiseen.

Työssä keskitytään verkkosovelluksen suunnitteluun ASP.NET-tekniikkaa käyttäen MVC-arkkitehtuurilla sekä C#-ohjelmointikielellä. Työ oli alun perin tarkoitus toteuttaa PHP-kielellä, mutta valitsin ASP.NETin, koska se ei ollut itselleni tuttu tekniikka, jolloin pystyn opiskelemaan työn edetessä uutta asiaa.

Tiedon lähteinä käytetään kahta eri kirjaa ja lukuisia Internet-lähteitä. Tietoa haetaan ohjelmistokomponenttien kotisivuilta, Microsoftin virallisilta sivuilta ja avoimesta tietosanakirjasta. Internet tarjoaa viimeisimmän tiedon monista tekniikoista, mutta tiedon oikeellisuudesta ei aina voida olla varmoja. Wikipedian kaltaisten avoimien tietosanakirjojen ongelma on niiden yhteisöllisyys. Kuka vain voi kirjoittaa mistä tahansa aiheesta mitä tahansa. Onneksi väärinkäyttöön puututaan Wikipediassa nopeasti. Wikipedian käyttöä lähteenä puolustaa tiedon valtava määrä. Taustatiedon hakemisessa Wikipedia toimii mielestäni hyvin.

Yritys, jolle verkkosivut tehdään, toimii Kouvolassa. Yritys suunnittelee ja valmistaa varjostimia sekä markkinoi tuotteitaan jälleenmyyjien kautta. Sivustosta ei ole tarkoitus toteuttaa verkkokauppaa, eikä sivustolla mainita varjostimien hintoja.

2 TEKNIIKAT

Tässä luvussa käsitellään sivuston toteutuksessa käytettyjä tekniikoita. Luvussa esitellään myös sivuston kehityksessä käytetty kehitystyökalu sekä tietokannan hallintajärjestelmä ja hallintatyökalu.

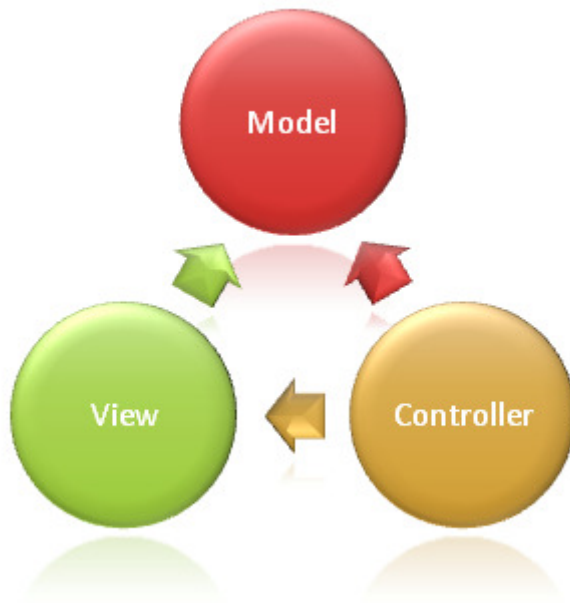
2.1 ASP.NET-verkkosovellus

ASP.NET (Active Server Pages .NET) on Microsoftin kehittämä palvelinperäinen verkkosivuteknologia. Espositon (2008) mukaan ASP.NET toimii muiden Web-kehitysympäristöjen tavoin HTTP-protokollan päällä ja käyttää hyväkseen HTTP:n komentoja ja käytäntöjä luodakseen kaksisuuntaisen selain-palvelinyhteyden. ASP.NETin erottaa erityisesti muista Web-kehitysteknologioista sen tarjoama abstrakti ohjelmointimalli, Web Forms.

ASP.NET-alusta tulee osana Microsoft .NET Frameworkia. Sovellukset koostuvat uudelleenkäytettävistä ja laajennettavista osista ja sovellusten käytössä on koko .NET Frameworkin luokkahierarkia. Sovellusten toteutuksessa voidaan käyttää tehokkaita ohjelmointikieliä, mukaan lukien: C#, Microsoft VB.NET ja J#. ASP.NET tarjoaa paljon tehokkaita työkaluja ja ominaisuuksia. Ajax-tekniikalla voidaan parantaa verkkosivun dynaamisuutta ja käyttäjäystävällisyyttä. (Esposito 2008.)

MVC-arkkitehtuuri

ASP.NET MVC (ASP.NET Model-View-Controller) on osa ASP.NET-verkkosovellusten kehitysympäristöä. Se on toinen kahdesta ohjelmointimallista, joita voidaan käyttää verkkosovellusta tehdessä. Toinen malli on Web Forms. MVC-sovellus toteutetaan jakamalla ohjelma kolmeen osaan (kuva 1).



Kuva 1 MVC-arkkitehtuurin osat (MS ASP.NET)

Malli (Model) huolehtii sovellusaluekohtaisen tiedon tallentamisesta, ylläpidosta ja käsittelystä. Näkymä (View) määrittää sovelluksen tietojen esitystavan ja ulkoasun. Verkkosovelluksessa näkymä sisältää tyypillisesti XHTML-merkkausta. Ohjain (Controller) vastaa sovelluksen ohjauslogiikasta. Se on vuorovaikutuksessa mallin ja näkymän kanssa sekä hallitsee sovelluksen tiedonkulkua ja suoritusta. Sovelluksen pilkkominen osiin tuo joustavuutta sovelluksen kehittämiseen. Esimerkiksi ulkoasu eli näkymä voidaan vaihtaa muuttamatta ohjainta, tai ohjainta voidaan muokata muuttamatta näkymää. Verkkosovellusta kehitettäessä tiiminä näkymiä voivat kehittää käyttöliittymäsuunnittelijat ja ohjaimia sekä malleja ohjelmistokehittäjät. (MS ASP.NET; MVC Wikipedia.)

2.2 XML

XML (eXtensible Markup Language) on yksinkertainen merkkauskieli, jolla voidaan kuvata tiedon merkitys tiedon sekaan. XML:n avulla voidaan jäsentää suuri tietomäärä selkeästi. XML ei ole sivunkuvauskieli, kuten HTML. XML on määritelty siten, että dokumentti voidaan helposti käsitellä tietokoneohjelmalla, tuntemalla ainoastaan kielen muodolliset säännöt. XML-dokumentissa tiedetään aina alku- ja lopputagien perusteella jokaisen elementin alku- ja loppukohta.

XML-dokumenttien käsittelyyn on laajalti työkaluja. (Korpela 2004; XML Wikipedia.)

Esimerkki XML-dokumentista:

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<asiakas>
  <asiakas-id>100</asiakas-id>
  <etunimi>Antti</etunimi>
  <sukunimi>Asiakas</sukunimi>
</asiakas>
```

Dokumentti alkaa Korpelan (2004) mukaan prologilla (prolog), joka kertoo dokumentin sisältävän XML-määrittelyn mukaista tietoa. Prologi voi myös sisältää kuvauksen dokumenttityypistä, tai viittauksen siihen. Prologin voi jättää pois, jos dokumenttia käsittelevä ohjelma tietää muulla perusteella käsittelevänsä XML-dokumenttia.

Dokumentin ensimmäinen rivi on XML-ilmoitus (XML declaration). Siinä ilmoitetaan dokumentin määrittelyn versio ja käytetty merkkikoodaus. Muita mahdollisia merkkikoodauksia ovat esimerkiksi utf-8 ja utf-16. Esimerkissä juurielementtinä on asiakas. Juurielementin sisällä on muita elementtejä, jotka sisältävät tiedot asiakkaasta. (Korpela 2004.)

Muotoilusäännöt

XML:ssa voidaan määrittää tarkasti, miten elementtejä sallitaan sisäkkäin. Tämä tarkoittaa dokumenttityypin määrittelyä, jolla kuvataan dokumentin merkkauksen muodollinen rakenne, eli syntaksi. Dokumenttien syntaksin ohjelmallisessa tarkistuksessa tarvitaan muotoilusääntöjä. Muototarkistus on tärkeä, koska XML-dokumentteja käsittelevät ohjelmat ja selaimet eivät normaalisti osaa käsitellä muotovirheitä dokumenteissa. Merkkauksessa käytettyjen elementtien merkitys voidaan määritellä erikseen sanallisesti. Näin täytyykin toimia, jos merkkauksielestä tahdotaan yleiskäyttöinen. (Korpela 2004.)

Oikeellisuus

XML-dokumentin täytyy Wikipedian mukaan noudattaa vähintään seuraavia sääntöjä, jotta se olisi hyvin muodostettu ja syntaksiltaan oikea:

- Dokumentissa on yksi juurielementti.
- Ei-tyhjillä elementeillä on oltava alku- ja lopputagi. Tyhjät elementit merkitään erikseen.
- Attribuutit merkitään lainausmerkkien sisään.
- Sisäkkäisten elementtien alku- ja lopputagit eivät mene ristiin toistensa kanssa.

XML-dokumenttia voidaan pitää validina, jos se on jonkin dokumenttityypin mukainen. (XML Wikipedia.)

Yleinen pohja

XML-kielellä on tärkeä merkitys senkin vuoksi, että se on laajalti käytetty yhteinen perusta monille merkkaukielille ja tiedostomuodoille. XML on monikäyttöinen yksinkertaisuutensa vuoksi, ja se soveltuu hyvin tiedonsiirtoon sovellusten välillä. Monet sovellukset osaavat oman tiedostomuotonsa lisäksi käsitellä XML-muodossa olevia tiedostoja. (Korpela 2004.)

2.3 XHTML

XHTML (eXtensible Hypertext Markup Language) on Korpelan mukaan yleisnimitys HTML:n uusille versioille. X-kirjain johtuu sanasta eXtensible, joka tarkoittaa laajennettavissa olevaa. XHTML ei itsessään ole kuitenkaan eri tavoin laajennettavissa kuin perinteinen HTML. X-kirjain viittaa enemmänkin XML:ään. XHTML noudattaa XML-kielen muotoilusääntöjä. XHTML on siis yhteensopiva XML-sovellusten kanssa.

Web-sivujen toteutus XHTML-kielellä vaatii suurempaa tarkkuutta ja järjestelmällisyyttä verrattuna HTML:ään. XHTML:ssä elementtien nimet kirjoitetaan

pienin kirjaimin, elementeillä on oltava alku- ja lopputagit (kuten XML:ssä) ja määritteiden arvot on kirjoitettava lainausmerkkeihin. XHTML:ään liittyy myös muita kielelle ominaisia merkkauksen erityispiirteitä, joita ei tässä käsitellä. (Korpela 2004.)

2.4 CSS

Tyyliohjeiden (style sheet) tarkoitus on erottaa dokumentin ulkoasun määrittely sen sisältöä kuvaavasta rakenteesta. Käytännössä tyyliohje koskee sivun graafista esitystapaa selaimessa tai paperilla. Tyyliohjeiden kielenä käytetään CSS:ää (Cascading Style Sheets). Kielestä on käytössä kaksi versiota, CSS 1 ja huomattavasti laajempi CSS 2. Kolmas versio CSS 3 on ollut jo pitkään kehitteillä. (Korpela 2004.)

Säännöt ja syntaksi

CSS-kielinen tyyliohje koostuu Wikipedian mukaan säännöistä. Yhden säännön perussyntaksi on seuraavanlainen:

```
valitsin { ominaisuus: Arvo; }
```

Käytetään esimerkkinä seuraavaa tyyliohjetta:

```
h1, h2 { color: #ffffff; text-decoration:underline; }
```

Valitsimet h1 ja h2 kohdistavat säännön kyseisiin otsikkoelementteihin. Aaltosulkeiden välissä ovat deklaraatiot, jotka erotetaan toisistaan puolipisteellä. Deklaraatiolla asetetaan elementin ominaisuuden arvo. Esimerkissä määrätään siis, että kyseisten otsikkoelementtien teksti on väritään valkoista ja alleviivattua. Tyyliohjeessa voi olla sääntöjen seassa kommentteja. Kommentti alkaa merkeillä /* ja loppuu merkkeihin */. (Korpela 2004; CSS Wikipedia.)

Esimerkki kommentin käytöstä:

```
h1 { color: #ffffff; /* Otsikon väri on valkoinen. */ }
```

Tyylin asettamisen vaihtoehdot

Tyylit voidaan asettaa elementeille Korpelan mukaan eri tavoilla. Oletetaan esimerkiksi, että aiemmin mainitulle h1-elementille halutaan asettaa musta tekstin väri. Tyyliohje voi olla HTML-dokumentissa elementin style-määritteessä:

```
<h1 style="color: #000000;">
```

Tyyliohje voi olla style-elementissä, joka on HTML-dokumentin head-osassa:

```
<style type="text/css">
  h1 { color: #000000; }
</style>
```

Tyyliohje on erillinen tiedosto tyyli.css, johon viitataan link-elementissä HTML-dokumentin head-osassa:

```
<link rel="stylesheet" href="tyyli.css" />
```

Tyyli.css-tiedoston sisältönä on:

```
h1 { color: #000000; }
```

Näistä kolmesta eri tavasta kaksi ensimmäistä esittävät sisäisen tyyliohjeen käyttämistä. Kolmas tapa käyttää ulkoista tyyliohjetta, eli erillistä tiedostoa. Minäkään muun tyyliohjeen ollessa vaikuttamatta esitystapaan, kaikki kolme antavat saman lopputuloksen. Eri tyyliohjeet saattavat vaikuttaa samojen elementtien tyyliin. Tällöin tyyliohjeiden esitystapa voi olla merkitsevä sen suhteen, mikä kilpailevista tyyleistä jää voimaan. Jos esimerkiksi elementin samaan ominaisuuteen kohdistuu sääntö ulkoisessa tyyliohjeessa ja elementin style-määritteessä, sisäinen tyyliohje jää voimaan. Tämä mahdollistaa esimerkiksi valikkojen muotoilun muuttamisen dynaamisesti riippuen siitä, missä osassa sivustoa käyttäjä milläkin hetkellä on. (Korpela 2004.)

2.5 Visual Studio

Visual Studio on ohjelmistojen kehitysympäristö, joka sisältää monta eri visuaalista suunnittelu- ja kehitystyökalua. Kehitysympäristöllä voidaan tuottaa muun muassa Windows-, verkko- ja matkapuhelinsovelluksia. Sovellusten tuottamiseen vaadittavat kirjastot, tietokannat, kansiot ja tiedostot ovat ryhmitelty kahdelle tasolle, jotka kehitysympäristössä esiintyvät nimillä solutions eli ratkaisut, ja projects eli projektit. Ratkaisu on suurempi kokonaisuus, joka voi sisältää monta projektia. Projekti sisältää kehitettävän ohjelmiston vaatimia tietoja ja tiedostoja. Näiden kahden tason ansiosta kehitettävän ohjelmiston asetuksia voidaan muuttaa kaikille projekteille kerralla tai yksitellen jokaiselle projektille. (Esposito 2008.)

Kehitysympäristöllä voidaan tuottaa kokonainen verkkosivusto ilman Internet-yhteyttä. Tietokantatukea varten paikallisella koneella täytyy olla asennettuna SQL Server -tietokantajärjestelmä (ks. luku 2.6). Kehitysympäristössä on sisäänrakennettuna ASP.NET-tekniikkaa tukeva palvelin. Palvelimen ansiosta sivusto voidaan kehittää täysin valmiiksi, ennen sivuston julkaisua tai siirtämistä Internetiin. Verkkosivusto voidaan julkaista paikalliselle koneelle haluttuun kansioon tai suoraan Internetissä olevalle palvelimelle. Julkaisuvaiheessa haluttuun sijaintiin kopioidaan vain sivuston vaatimat tiedostot eikä mitään ylimääräistä. Paikalliselle koneelle julkaistaessa sivusto voidaan siirtää Internetiin FTP:n (File Transfer Protocol) avulla erillisellä ohjelmalla.

2.6 SQL

SQL (Structured Query Language) on standardoitu rakenteinen kyselykieli, jolla voidaan hakea ja muuttaa tietokannan tietoja, sekä muuttaa tietokannan rakennetta ja oikeuksia. SQL on standardi käsiteltäessä ja haettaessa tietoja tietokannasta. SQL:ää hyödynnetään muissa tietokantoja käsittelevissä tekniikoissa ja ratkaisuissa, kuten esimerkiksi LINQ to SQL -komponentissa (ks. luku 2.7). Yksi SQL:ää hyödyntävä järjestelmä on Microsoft SQL Server-tietokantajärjestelmä, joka käsitellään myöhemmin tässä luvussa. (SQL 2Kmediat.)

Taulut

Relaatiotietokantajärjestelmissä tieto esitetään Gertzin (2000) mukaan tauluja käyttämällä. Tietokantaan lähetettävän kyselyn tuloksena saadaan myös taulu. Taulu koostuu riveistä ja sarakkeista (taulukko 1).

Taulukko 1 Tietokannan taulun rakenne

	Sarake 1	...	Sarake n
Rivi 1			
...			
Rivi n			

Taulun nimi on yksilöllinen. Taulu sisältää rivejä, jotka sisältävät tauluun tallennetun tiedon. Taululla voi olla yksi tai useampi sarake. Sarake koostuu sen nimestä ja datatyypistä. Datatyypit kuvaavat rivien ominaisuudet. Ominaisuudet määräävät taulun rakenteen, eli taulun relaatiomallin. Datatyyppien perusteella määräytyy, minkä tyyppistä informaatiota tauluun voidaan tallentaa. (Gertz 2000.)

Kyselyt

Kyselyiden avulla muokataan tai haetaan tietokannan tietoja. Kyselykielenä toimii SQL. SQL-kyselyn yleisrakenne on Laineen (2000) mukaan seuraava:

```
select <tulostietomäärittely>  
from <kohdetaulut>  
[where <valintaehdot>]  
[group by <ryhmitystekijät>]  
[having <ryhmärajoitteet>]  
[order by <järjestysperusta>]
```

Hakasuluissa olevat komponentit ovat vaihtoehtoisia. Tulostietomäärittelyssä määritellään tulosjoukon sarakkeet. Kohdetaulut ovat taulut, joihin kysely kohdistetaan. Valintaehdot ovat ehtoja, joiden perusteella tulostaulun rivit määräytyvät. Ryhmitystekijöillä määrätään tulostaulu ryhmäperustaiseksi.

Ryhmärajoitteilla valitaan halutut ryhmät. Järjestysperustan avulla voidaan järjestää tulostaulun rivit sisällön perusteella. (Laine 2000.)

Microsoft SQL Server Express Edition

Microsoft SQL Server Express Edition, lyhyemmin SQL Server Express, on Microsoftin kehittämä relaatiotietokannan hallintajärjestelmä. Järjestelmällä hallitaan tiedonsiirtoa asiakkaan ja tietokannan välillä. Järjestelmä vastaanottaa pyynnöt asiakkaalta, suorittaa pyynnön tietokannassa ja palauttaa vastauksen asiakkaalle. (Myhosting.)

SQL Server Express integroituu Visual Studioon, mikä helpottaa ja nopeuttaa sovellusten kehittämistä. SQL Server Express voi toimia tietokantajärjestelmänä kehittäjän järjestelmässä tai palvelimessa. Express Edition -sanat tuotenimen perässä tarkoittavat maksutonta versiota kyseisestä tuotteesta. (SQL Server Express Readme Microsoft.)

Microsoft SQL Server Management Studio Express

Microsoft SQL Server Management Studio Express on ilmainen graafinen hallintatyökalu SQL Server Express -järjestelmän tarjoamien tietokantojen hallintaan (SSMSE Microsoft). Työkalun avulla voidaan hallita tietokantojen ominaisuuksia. Seuraavassa listassa on lueteltu työkalun tärkeimpiä toimintoja:

- paikalliseen tai etätietokantaan yhdistäminen
- tietokantojen luonti
- taulujen luonti
- taulujen rakenteen muokkaus
- taulujen sisällön muokkaus
- tietokantojen varmuuskopiointi ja palauttaminen
- kyselyiden suorittaminen tietokannassa
- raporttien luonti tietokannasta.

2.7 LINQ to SQL

LINQ (Language Integrated Query) on Järvisen (2008) mukaan Visual Studio 2008 -versiossa käytettävissä oleva tekniikka, joka yhdistää ohjelmointi- ja kyselykielet. LINQ tarjoaa .NET-sovelluksille mahdollisuuden hakea tietokannan tietoja ohjelmointikielen omilla komennoilla. LINQ:n avulla kyselyjä voidaan tehdä tietokantojen lisäksi olioihin, kuten taulukot ja kokoelmat, ja XML-tiedostoihin. Kyselyn syntaksi on tietolähteestä riippumatta lähes sama, joten kehittäjän ei ole välttämätöntä tietää tietolähdettä. Esimerkkinä käytetään Tietokone-lehden artikkelissa (Järvinen 2008) olevaa ohjelmakoodia, joka hakee kokonaislukutaulukosta parilliset luvut nousevasti järjestettynä:

```
int[] numerot = { 4, 8, 12, 7, 3, 5, 10, 2, 9, 11, 6 };

var parilliset = from n in numerot
where n % 2 == 1
orderby n ascending
select n;

foreach (int parillinen in parilliset)
{
    MessageBox.Show(parillinen.ToString());
}
```

LINQ to SQL on Visual Studion komponentti, jonka avulla relaatiotietokannan tietoja voidaan käsitellä olioina. Syntaksi muistuttaa joiltakin osin SQL-kieltä, sillä siitä löytyy tuttuja avainsanoja kuten: select, where ja orderby. (Järvinen 2008.)

LINQ to SQL kääntää kieleen integroidut kyselyt SQL-kyselyiksi suoritettavaksi tietokannassa. Tämän jälkeen tekniikka kääntää kyselyn vastauksen eli tulostaulun ohjelmakoodissa määrätyksi olioksi. Olion tietoja voidaan tämän jälkeen hyödyntää ohjelmassa. (LINQ to SQL MSDN.)

Otetaan esimerkkinä yhden rivin hakeminen ja tulostaminen tietokannasta C#-kieltä käyttäen:

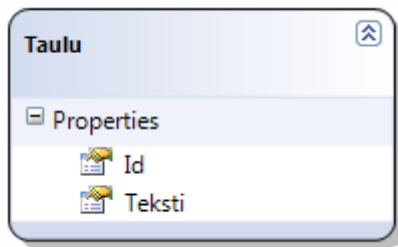
```

var db = new
DataContext(@"server=.\SQLEXPRESS;database=Testikanta;integrated
security=SSPI");

var yksi = db.Taulus.SingleOrDefault(x => x.Id == 3);
console.WriteLine(yksi.Teksti);

```

Esimerkissä on käytössä tietokanta nimeltä Testikanta, joka sisältää yhden taulun nimeltä Taulu (kuva 2). Taululla on kaksi saraketta: Id, joka sisältää rivin yksilöivän numeron, sekä Teksti, johon voidaan tallentaa tekstimuodossa olevaa tietoa.



Kuva 2 Esimerkin taulu

Ensimmäinen rivi luo yhteyden tietokantaan, jotta sitä voidaan käsitellä muuttujan db avulla. Muuttujaan yksi haetaan taulusta yksi rivi. Hakuehtona on, että kyseisen rivin yksilöivän numeron täytyy olla 3. Haun jälkeen tulostetaan rivin Teksti-sarakkeen arvo.

Tarkastellaan toista esimerkkiä, jossa käytetään samaa tietokantaa. Esimerkissä haetaan ja tulostetaan kaikki taulun rivit, joiden Teksti-sarakkeen arvossa esiintyy a-kirjain. Tietokantaan yhdistävä rivi on sama, mutta se on jätetty pois tilan säästämiseksi.

```

var monta = from taulu in db.Taulus
            where taulu.Teksti.Contains("a")
            select taulu;

foreach (var rivi in monta)
    console.WriteLine(rivi.Teksti);

```

2.8 jQuery

jQuery on Wikipedian mukaan kaikkia selaimia tukeva JavaScript-kirjasto, joka parantaa ja yksinkertaistaa JavaScriptin ja HTML:n välistä vuorovaikutusta. Kirjaston avulla voidaan tehdä joustavampia ja tehokkaampia verkkosivuja. jQuery on ilmainen avoimen lähdekoodin ohjelmisto.

Käyttäjät voivat tehdä kirjastoon laajennuksia ja tällä tavalla lisätä kirjaston toiminnallisuutta. Laajennusten avulla kirjaston etuja voidaan hyödyntää entistä paremmin ja verkkosivuista saadaan dynaamisemmat. Kirjaston syntaksi kehitettiin helpottamaan dokumentin navigointia, Document Object Model (DOM) -elementtien käsittelyä, animaatioiden luontia, tapahtumien käsittelyä ja Ajax-sovellusten tuottamista. Kirjasto sisältää seuraavia ominaisuuksia:

- DOM-elementtien valitsimet
- DOM läpikäynti ja muokkaaminen
- tapahtumien käsittely
- CSS-tyylien muokkaaminen
- tehosteet ja animaatiot
- Ajax
- laajennettavuus.

Kirjaston uusin versio voidaan ottaa käyttöön verkkosivulla lisäämällä sivun merkkaukseen seuraava merkintä:

```
<script src="http://ajax.microsoft.com/ajax/jquery/jquery-1.4.2.min.js" type="text/javascript"></script>
```

Tämän jälkeen kirjaston ominaisuuksia voidaan hyödyntää sivulla script-lohkojen sisällä. (jQuery Wikipedia.)

3 SUUNNITTELU

Luvussa perehdytään verkkosovelluksen suunnittelun ja toteutuksen vaiheisiin. Suunnittelussa ainut ennalta määrätty asia oli malli sivun tulevasta ulkoasusta. Ulkoasua lukuun ottamatta sivuston suunnittelu oli täysin vapaamuotoista. Sivuston sisällöstä haluttiin tehdä muokattava. Tätä varten tehtiin hallintaosio, josta kerrotaan lisää myöhemmin tässä luvussa.

3.1 Sivuston ulkoasu

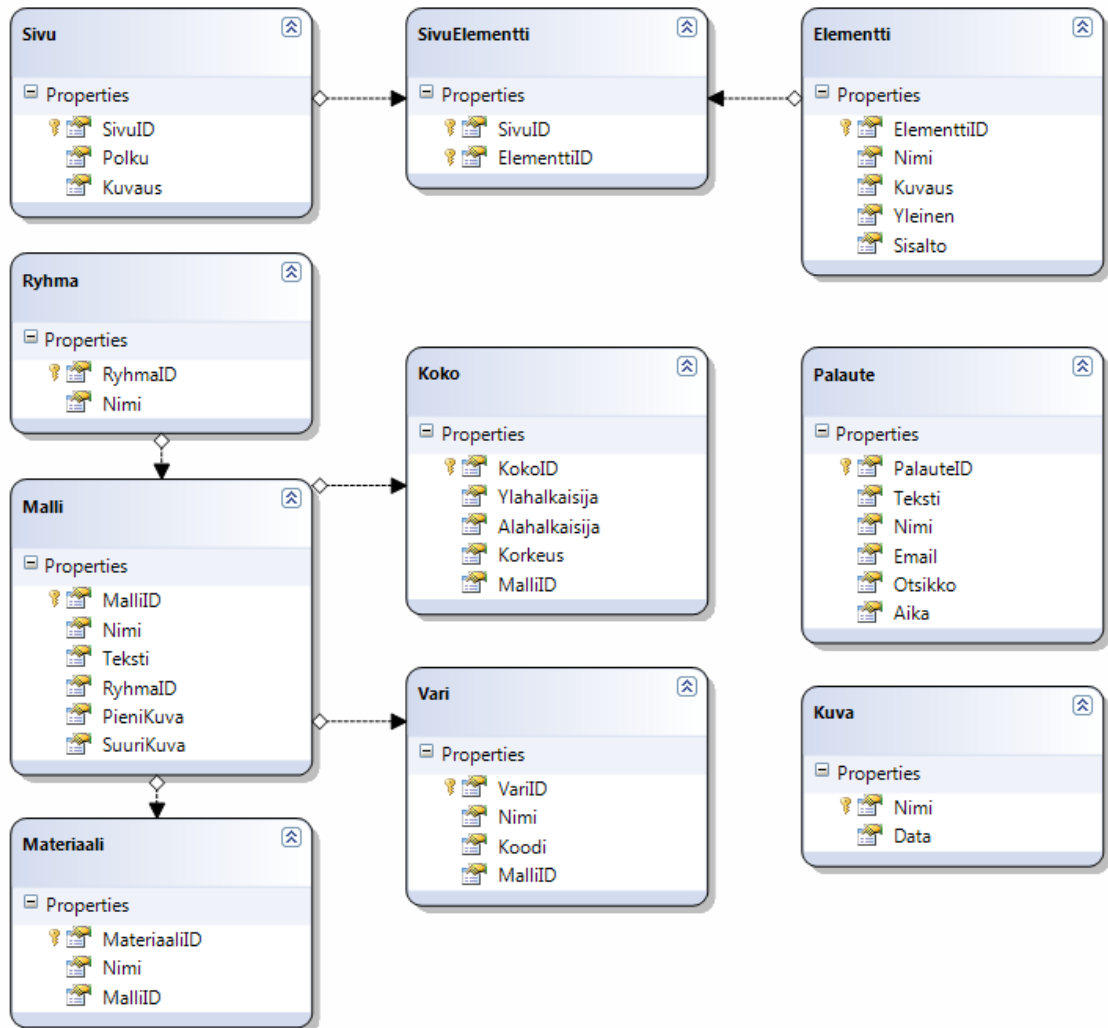
Ensimmäinen asia sivun toteutuksessa oli ulkoasun suunnittelu ja luominen. Aluksi suunniteltiin sivupohjana käytetty kuva, animoitu logo ja päälinkkien kuvat. Suunnittelu aloitettiin ulkoasusta, koska tämän jälkeen sivustoa on helppo käydä rakentamaan ulkoasuun sopivaksi ja elementit saadaan sijoiteltua helpommin oikeille paikoilleen. Ulkoasuun liittyy kuvien lisäksi olennaisesti myös sivuston tyylitiedosto, jota kehitettiin jatkuvasti sivustoa rakennettaessa. Ulkoasuun liittyviä tärkeimpiä asioita ovat selkeys, helppokäyttöisyys ja yhteensopi- vuus eri selaimien kanssa. Ulkoasusta saatiin selkeä ja sen toimivuus varmistettiin Internet Explorer 6- ja 7- sekä Mozilla Firefox -selaimilla.

3.2 Tietokantarakenne

Ulkoasun suunnittelun jälkeen alettiin suunnitella sivuston tietotarpeita. Tietotarpeita mietittiin ja hahmoteltiin paperille. Tietotarpeista keskusteltiin asiakkaan kanssa ja mietittiin, mitä tietoja varjostinmalleille tulisi voida tallentaa. Tietokannan taulujen ja yhteyksien suunnitteluun käytettiin Visual Studion sisäänrakennettua graafista työkalua.

Tietokannan suunnittelussa piti ottaa huomioon sisällönhallinnan, mallien hallinnan ja palautteiden hallinnan tietotarpeet. Käyttäjien hallintaa ei sivustoon toteutettu. Hallintaosiota varten tehtiin yksi sisäänrakennettu käyttäjä. Tietokantaa ei saatu suunniteltua ensimmäisellä kerralla sovelluksen ohjelmoinnin kannalta järkeväksi, joten toteutuksen edetessä tietokantaan tehtiin pieniä muutoksia.

Muutokset koskivat taulujen välisten suhteiden muuttamista ja kahden välirelaation poistamista. Lopullinen tietokanta on selkeä ja toimiva (kuva 3).



Kuva 3 Tietokantarakenne

Tietokannan taulut Sivu, SivuElementti ja Elementti liittyvät sisällönhallintaan ja sisältävät muun muassa sivuston päälinkkien alta löytyvät tiedot. Koska sivulla on useasti monta elementtiä ja elementti voi esiintyä monella sivulla, taulujen välille tarvittiin monta-moneen-yhteys. Kuva-taulu tarvitaan kuvien tallentamiseksi tietokantaan. Tietokantaan tallennettuja kuvia voidaan lisätä sisällönhallinnan kautta näytettäväksi halutulla sivulla. Mallien hallintaan liittyvät taulut Ryhma, Malli, Koko, Materiaali ja Vari sisältävät varjostinmallien tarvitsemat tiedot. Palaute-tauluun tallennetaan vierailijoiden lähettämät palautteet.

3.3 Sivuston hallintaosio

Hallintaosion suunnittelu ja toteutus vei suurimman osan ajasta. Aluksi hallinnan toimintoja hahmoteltiin julkisen osion sivupohjan päälle. Pian kuitenkin huomattiin, että kyseinen sivupohja oli liian ahdas käytettäväksi hallintaa varten. Sivuston hallinnalle jouduttiin siis suunnittelemaan oma tyyli ja grafiikka.

Hallintaosiossa käytettiin jQuery-kirjastoa lomakkeen kenttien syötteiden oikeellisuuden tarkistukseen. Tekniikkaa käytettiin myös tietojen siirtämiseen lomakkeen eri kenttien välillä, esimerkiksi lisättäessä kokoja, värejä tai materiaaleja mallille eri valintaluetteloihin.

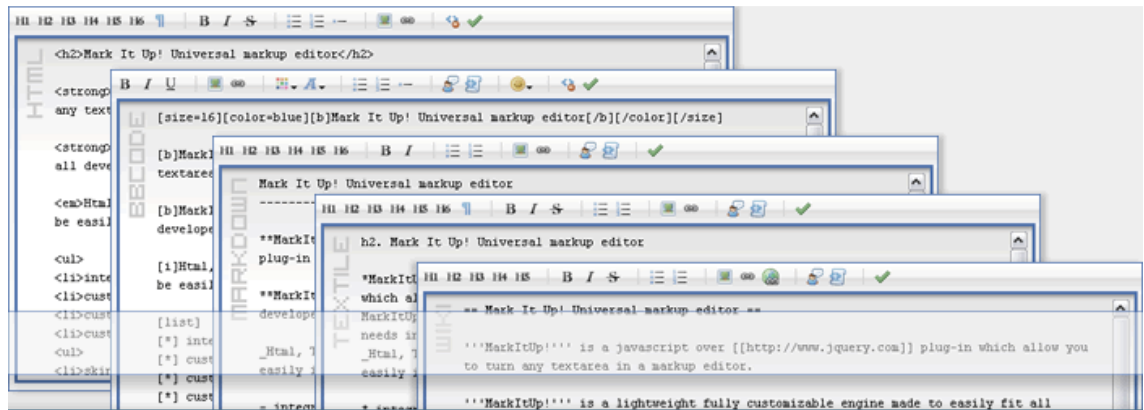
Hallinnassa käytettiin hyödyksi valmiita komponentteja. iColorPicker on jQuery:llä toteutettu verkkosivulle lisättävä värivalitsin. Se on alun perin tarkoitettu lisättäväksi tekstikenttään, jolloin värin valinnan yhteydessä muutetaan tekstikentän taustaväri ja asetetaan kentän arvoksi valitun värin koodi (kuva 4).



Kuva 4 iColorPicker-komponentti (SuperSite)

Komponenttia käytetään hyödyksi varjostinmallien värien määrittämistä varten. Mallia muokattaessa voidaan mahdolliset väri vaihtoehdot määrittää valitsemalla halutut värit ja kirjoittamalla väreille nimet. Komponentin käyttäytymistä jouduttiin muuttamaan, sillä se ei alkuperäisen toteutuksen mukaan ole jatkuvasti näkyvillä vaan avautuu tekstikentän vieressä olevasta painikkeesta.

markItUp! on monia eri merkintäkieliä tukeva verkkosivulle liitettävä editori. Se muuttaa normaalin tekstikehyksen editoriksi, lisäämällä siihen pikanappuloita ja esikatselumahdollisuuden (kuva 5).



Kuva 5 markItUp!-komponentti (Salvat 2010)

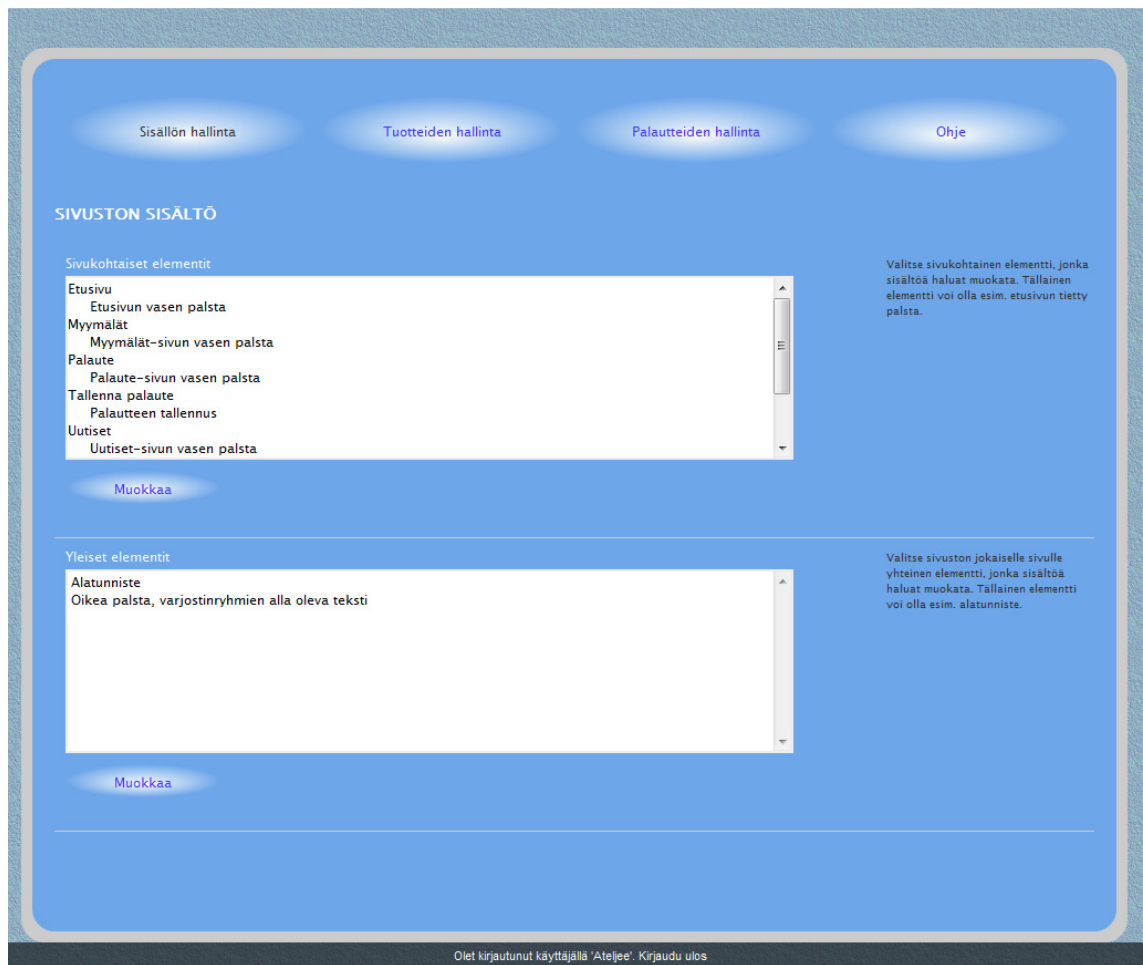
Tässä tapauksessa editorin merkintäkieleksi määriteltiin HTML. Editorin perusasetuksia ei suuremmin muutettu. Editorin toiminnot suomennettiin ja esikatselunäkymää muokattiin, jotta se näyttäisi ulkoasultaan samalta kuin sivuston julkisivu. Esikatselunäkymästä tehtiin samankokoinen julkisivun pääpalstan kanssa, joten sisältöä muokattaessa nähdään, miten sisältö asettuu kyseiseen tilaan.

Ylläpitotoiminnot

Hallintaosion ylälaudassa on neljä päälinkkiä navigointiin: sisällön hallinta, tuotteiden hallinta, palautteiden hallinta ja ohje. Kyseisistä toiminnoista voidaan hallita sivustolla näkyvää sisältöä, varjostinmalleja ja vierailijoiden palautteita. Ohje-sivulta nähdään verkkosovellukseen liittyvää tietoa.

Sisällönhallinta

Sisällönhallinnassa voidaan tarkastella sivuston elementtejä. Elementit ovat sivuston sivujen osia, joiden sisältöä voidaan muokata. Elementtejä on kahta tyyppiä: sivukohtaiset ja yleiset elementit. Sivukohtainen elementti sijaitsee vain yhdellä sivulla tietyssä kohdassa. Yleinen elementti sijaitsee kaikilla sivuilla tietyssä kohdassa, kuten alatunniste. Hallintasivulla listataan erityyppiset elementit eri listoihin (kuva 6).



Kuva 6 Sisällön hallinta

Elementtiä voidaan muokata valitsemalla se valintalistasta ja painamalla Muokkaa-painiketta. Painikkeen painamisen jälkeen avautuu Elementin muokkausnäky (kuva 7).

Sisällönhallintaa voidaan jatkokehittää toteuttamalla sivujen ja elementtien lisäysmahdollisuus. Tällä hetkellä on vain mahdollista muokata ennalta määrättyjä elementtejä. Tarkoituksena onkin tulevaisuudessa kehittää kyseistä sisällönhallintaa lisäten siihen ominaisuuksia ja tehdä siitä itsenäinen kokonaisuus, joka on helppo liittää uusiin verkkosovelluksiin sisällönhallintatyökaluksi.

Sisällön hallinta








































Tuotteiden hallinta

Palautteiden hallinta

Ohje

ELEMENTIN MUOKKAUS – MYYMÄLÄT-SIVUN VASEN PALSTA

Sisältö

H1 H2 H3 H4 H5 H6 **B** *I* ~~S~~                                       

Kuvan näkymässä ollaan muokkaamassa Myymälät-sivun vasen palsta-elementtiä. Elementin sisältö näkyy julkisivulla Valaisin- ja varjostinmyymälät-päälinkkiä painettaessa. Tekstilaatikossa näkyy elementin sisältämä XHTML-koodi. Tekstilaatikon alla oleva esikatseluruutu ei ole näkymään tullessa automaattisesti näkyvässä, vaan se täytyy avata työkalurivin oikeassa laidassa olevasta Esikatselu-painikkeesta. Esikatseluruutu käyttää samaa tyyliohjetta kuin sivuston julkisivu, joten muokausvaiheessa nähdään samalla tavalla muotoiltu elementti, kuin miltä se näyttää sivuston vierailijoille. Tämä helpottaa halutun lopputuloksen saavuttamista muokausvaiheessa.

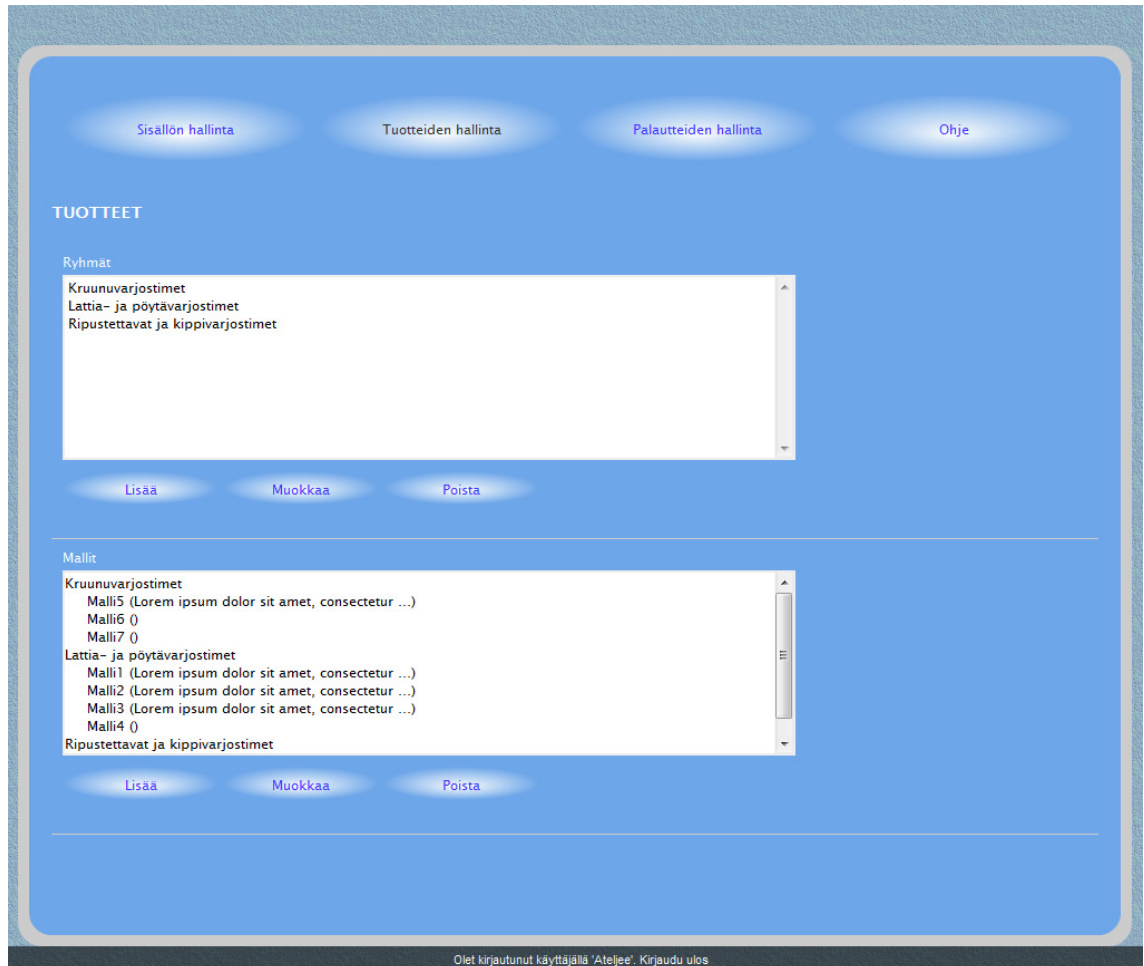
Työkaluriviltä löytyvät seuraavat HTML-tagit: otsikot (h1-h6), kappale (p), lihavointi (strong), kursivointi (em), yliviivaus (del), listat (ul, ol, li), kuva (img) ja linkki (href) sekä toimintopainikkeet siivous ja esikatselu. Työkalurivin painiketta painettaessa kyseinen HTML-koodi lisätään kohdistimen osoittamaan kohtaan tekstilaatikossa.

Esikatseluruudun alapuolella listataan tietokantaan tallennetut kuvat. Kuvat listataan tiedostonimien perusteella. Tiedostonimet ovat linkkejä, joista painettaessa kyseisen kuvan viittaus lisätään tekstilaatikkoon kuvan osoittamalla tavalla. Kyseisellä tavalla HTML-koodin sekaan tallennetut kuvaviittaukset muutetaan kuviksi sivun muodostusvaiheessa.

Tallennettujen kuvien listan alapuolelta voidaan tallentaa uusia kuvia tietokantaan. Haluttu kuva haetaan Selaa-painikkeella ja tallennetaan alla olevalla Tallenna-painikkeella. Tallentaminen ja kuvien listaus tehdään Ajaxin avulla. Tallennuksen jälkeen uusi kuva nähdään heti listassa. Kuvan lisäys ei onnistu, mikäli samanniminen kuvatiedosto löytyy jo tietokannasta. Tämä koskee vain sisällönhallinnan kuvia, eikä liity varjostinmallien kuvatiedostoihin. Kuvan tallennuksen onnistumisesta näytetään käyttäjälle ilmoitus. Kuvien poistoa ei ole toteutettu, mutta tämä ei ole haitaksi, sillä palveluntarjoaja ei ole rajoittanut tietokannan tallennuskapasiteettia. Elementin sisältö voidaan haluttujen muokkausten jälkeen tallentaa esikatseluruudun alla olevalla Tallenna-painikkeella.

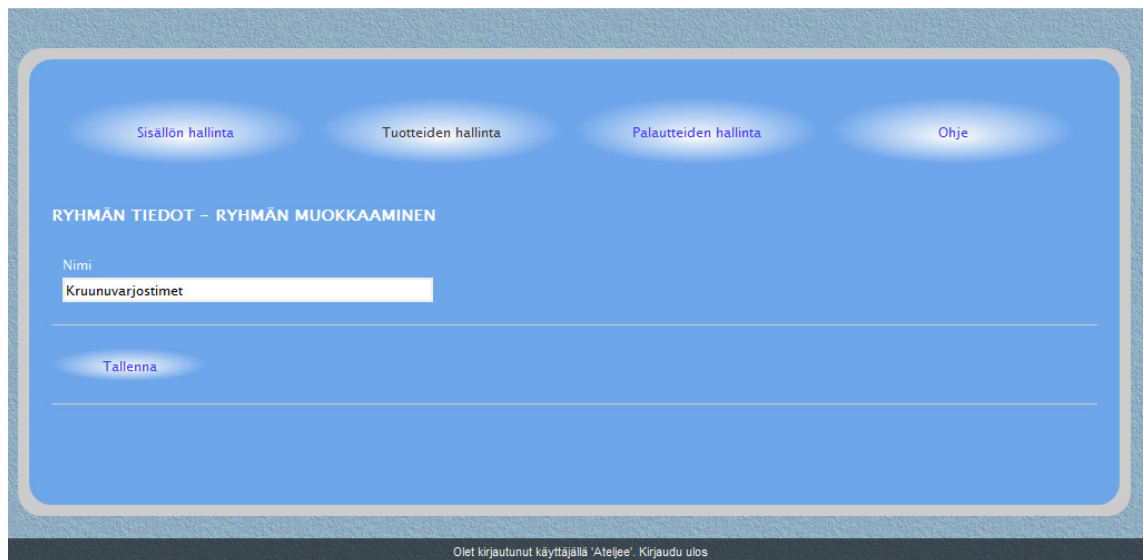
Tuotteiden hallinta

Tuotteiden hallintasivulla listataan varjostinryhmät ja -mallit. Ryhmät ovat ylemmässä listassa ja mallit alemmassa (kuva 8).



Kuva 8 Tuotteiden hallinta

Ryhmiä ja varjostinmalleja voidaan lisätä, muokata ja poistaa kunkin listan alla olevista painikkeista. Ryhmän poistaminen ei onnistu, mikäli yksikin malli kuuluu kyseiseen ryhmään. Sekä ryhmiä että malleja poistettaessa kysytään poiston vahvistus. Ryhmää lisättäessä tai muokattaessa aukeaa Ryhmän tiedotnäkö (kuva 9). Mallia lisättäessä tai muokattaessa aukeaa Mallin tiedotnäkö (kuva 10).



Kuva 9 Ryhmän muokkaus

Kuvan näkymässä voidaan muuttaa ryhmän nimeä. Ryhmän lisäämiseen käytetään samaa näkymää, mutta tekstilaatikkoon ei haeta tietoa. Ryhmä tallennetaan Tallenna-painikkeesta.

[Sisällön hallinta](#)
 [Tuotteiden hallinta](#)
 [Palautteiden hallinta](#)
 [Ohje](#)

MALLIN TIEDOT – MALLIN MUOKKAAMINEN

Nimi:

Teksti:

Ryhmä:

SAATAVILLA OLEVAT KOOT

Valitut koot: <- Lisää <- Poista Kaikki koot: Määritä saatavilla olevat koot.

LISÄÄ KOKO

Ylähalk: Alahalk: Korkeus:

Lisää

KÄYTETYT MATERIAALIT

Valitut materiaalit: <- Lisää <- Poista Kaikki materiaalit: Määritä saatavilla olevat materiaalit.

LISÄÄ MATERIAALI

Nimi:


Lisää

SAATAVILLA OLEVAT VÄRIT

Valitut värit: <- Lisää <- Poista Kaikki värit: Määritä saatavilla olevat väri vaihtoehdot.

LISÄÄ VÄRI



Väri nimi: Valitse väri: Lisää valittuihin väreihin uusi väri.



Lisää

MALLIN KUVAT

Esikatselukuva: Suuri kuva: Valitse vain JPG-muotoisia kuvia.

Tallenna

Olet kirjautunut käyttäjällä 'Abeljee'. Kirjautu ulos

Kuva 10 Mallin muokkaus

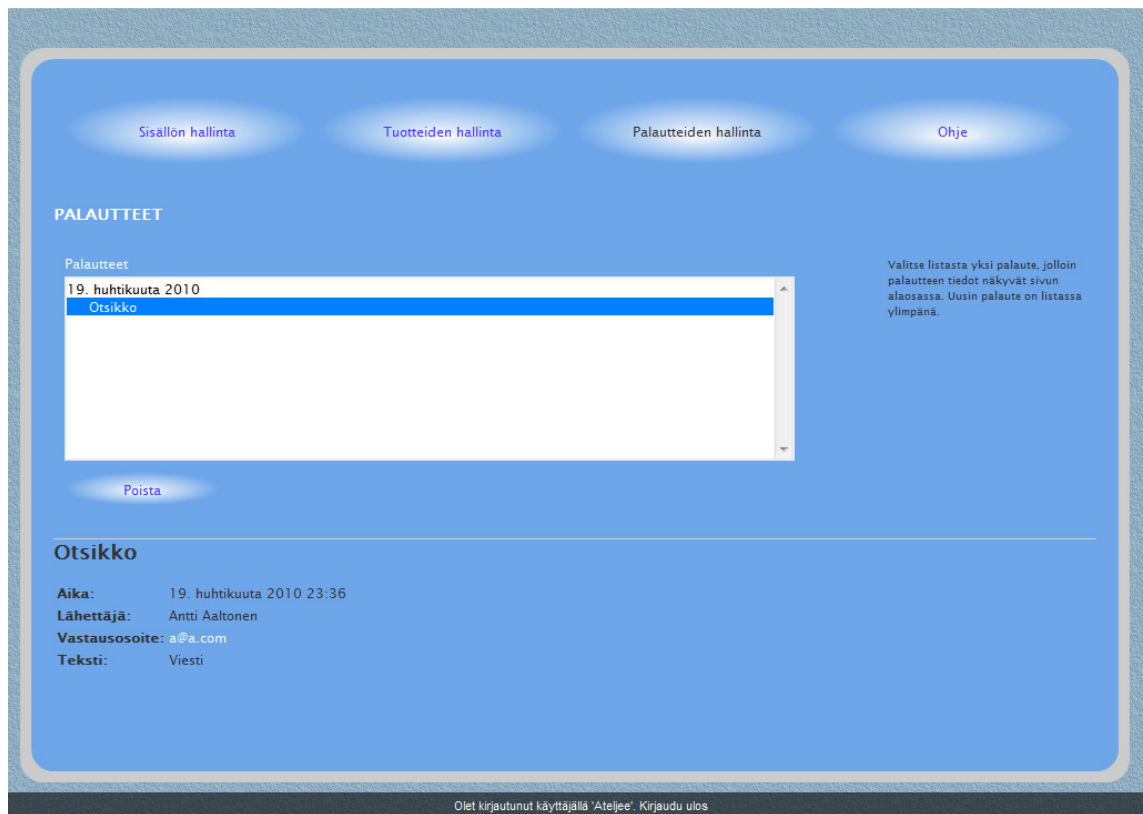
Kuvan näkymässä näytetään yhden mallin tiedot. Mallin lisäämiseen käytetään samaa näkymää, mutta tekstikentät ja valintalistat ovat oletuksena tyhjiä. Kuvassa näkyvät tiedot ovat vain täytteenä, ne eivät ole aitoja tietoja. Ensimmäiset kolme osiota sisältävät mallin nimen, kuvauksen ja ryhmän, johon malli kuuluu. Ryhmät haetaan tietokannasta ja niitä voidaan muokata tuotteiden hallinnassa.

Koko-, materiaali- ja värivalinnoissa vasemmanpuoleiset listat sisältävät tiedot, joita mallille on valittu. Oikeanpuoleiset listat sisältävät tiedot, joita on valittu aiemmin lisätyille malleille. Lisää-painikkeet kopioivat oikeanpuoleisesta listasta valitun tiedon vasemmanpuoleiseen listaan. Tämä vähentää samojen tietojen uudelleensyöttämistä, koska monia malleja valmistetaan samasta materiaalista ja malleja voi olla samankokoisia tai -värisiä. Poista-painikkeet poistavat valitun tiedon vasemmanpuoleisesta listasta. Uusia kokoja, materiaaleja ja värejä voidaan lisätä syöttämällä tiedot tekstikenttiin. Kokoa lisättäessä syötetään ylähalkaisija, alahalkaisija ja korkeus senttimetreinä omiin kenttiinsä. Materiaalia lisättäessä kenttään syötetään materiaalin nimi. Väriä lisättäessä kenttään syötetään värin nimi ja väripaletista valitaan kyseinen väri.

Viimeisenä mallille valitaan kuvat. Esikatselukuva näytetään julkisivulla malleja listatessa allekkain. Kuvatiedoston on suositeltavaa olla pienikokoinen, jotta suorituskyky ei kärsi malleja listatessa. Suuri kuva näytetään julkisivulla, kun tietyltä mallilta halutaan nähdä lisätiedot.

Palautteet

Palautteiden hallinnassa voidaan lukea ja poistaa vierailijoiden lähettämiä palautteita. Palautteiden otsikot näytetään valintalistassa päivän mukaan ryhmiteltyinä uusien ylimpänä (kuva 11).



Kuva 11 Palautteiden hallinta

Kun listasta valitaan yksi palaute, näytetään kyseisen palautteen tiedot sivun alaosassa. Palaute sisältää seuraavat tiedot: otsikko, lähetysaika, lähettäjän nimi, lähettäjän sähköposti ja viesti. Palautteita voidaan poistaa yksitellen valitsemalla haluttu palaute ja painamalla Poista-painiketta. Painikkeen painamisen jälkeen kysytään vielä poiston vahvistus.

Ohje

Ohje-sivulla näytetään yleisiä verkkosovellukseen liittyviä tietoja. Sivulla nähdään mm. sovelluksen versio, versiopäivämäärä, tuetut selaimet ja muut laitevaatimukset. Sivulta löytyy myös sähköpostiosoite yhteydenottoa varten mahdollisissa ongelmatapauksissa tai sovellukseen liittyvissä kysymyksissä.

3.4 Sivuston julkinen osio

Sivuston julkista osiota eli julkisivua kehitettiin alustavasti jo hallintaosion toteutuksen aikana, mutta pääasiassa vasta hallintaosion valmistuttua. Sivuston grafiikan toteutti kuvitukseen ja graafiseen suunnitteluun erikoistunut yritys, Grafi-lea. Graafiset elementit olivat pääpiirteissään valmiina, ennen kuin sivustoa alettiin muilta osin toteuttaa. Ulkoasua muutettiin hieman toteutuksen edetessä. Julkisivun ja sovelluksen suunnittelussa otettiin kantaa vain tekniseen toimivuuteen, vastuu sisällöstä jätettiin asiakkaalle.

Tarvittavat tiedot yhteydenottoon löytyvät Yhteystiedot-sivulta. Vaihtoehtoisesti voidaan käyttää Palaute-sivun lomaketta. Lomakkeeseen lisättiin validaattori, jolla tarkistetaan sähköpostiosoitteen kelvollisuus ja muihin kenttiin syötetyt tiedot. Näin vähennetään puutteellisten tai virheellisten tietojen tallentamista.

Kaikki julkisivun asiasisältö haetaan tietokannasta, ja sisältö on täysin ylläpitäjän muokattavissa. Jokainen selaimella avattava sivu sisältää elementtejä, joiden sisältöä voidaan muokata (ks. luku 3.3). Jokainen elementti voi sisältää mitä tahansa XHTML-merkkausta. Elementit sijaitsevat sivuilla ennalta määrättyissä paikoissa. Kuvassa nähdään etusivu, jossa muokattavia kohteita ovat vasemmalla sijaitseva palsta ja alatunniste (kuva 12).



Kuva 12 Julkisivu

Kuvassa nähdään sivuston ulkoasu ja yleinen rakenne. Päälinkit sijaitsevat vaakatasossa animoidun logon alla. Varjostinryhmät haetaan tietokannasta ja listataan linkkeinä allekkain oikeassa laidassa. Kaikki hallinnassa lisätyt ryhmät näkyvät myös julkisivulla. Vasen palsta sisältää vasta täytetietoja, lopullinen asiasisältö muokataan myöhemmin. Varjostinryhmän nimeä painamalla listataan ryhmään lisätyt varjostimet (kuva 13).

ATELJEE SINUN VARJOSI

Etusivu Uutiset Yritys Yhteystiedot Palaute Valaisin- ja varjostinmyymälät - valaisin valmistajat

Malli 1
 Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

[Lisätiedot ja suuri kuva](#)

Malli 2
 Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

[Lisätiedot ja suuri kuva](#)

Malli 3
 Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

[Lisätiedot ja suuri kuva](#)

Varjostimet
 Kruunuvarjostimet
 Lattia- ja pöytävarjostimet
 Ripustettavat ja kippivarjostimet

Seuraava sivu

© Ateljee Sinun Varjosi 2010

Kuva 13 Varjostinmallien listaus julkisivulla

Kuvassa listataan lattia- ja pöytävarjostimia. Malleilta puuttuvat vielä tarkat tiedot, ja tällä hetkellä tallennettuna on vasta mallien kuvia. Yhdellä kertaa näytetään maksimissaan kolme mallia aakkosjärjestyksessä. Jokaisesta mallista näytetään mallin nimi, kuvaus ja esikatselukuva. Muut tiedot saadaan näkyviin kuvien alla olevista linkeistä. Seuraava sivu -linkki näkyy ainoastaan mikäli tarpeellista, eli vain silloin, kun malleja riittää näytettäväksi seuraavallekin sivulle.

3.5 Julkaisu

Sivuston julkaisuvaihetta varten etsittiin ASP.NET-tekniikkaa ja MVC-arkkitehtuuria tukeva palveluntarjoaja. Mahdollisia vaihtoehtoja katsottiin alustavasti jo monta kuukautta aiemmin. Hyviä vaihtoehtoja löytyi monia. Monen palveluntarjoajan arvostelun lukemisen jälkeen valittiin Yhdysvalloissa toimiva yhtiö nimeltä MochaHost. Yhtiö oli saanut hyviä arvosteluja käyttäjiltä ja palvelimet tukevat monia uusimpia tekniikoita. Samassa sivutilassa on myös mahdollisuus pitää useampia verkkosivustoja.

Ensivaikutelma

Ensivaikutelma palveluntarjoajasta on hyvä. Maksun suorituksen jälkeen sivutila on käytössä muutamassa tunnissa. Sivustojen hallintaan käytettävä ohjauspaneeli on laaja, mutta tarjoaa selkeän käyttöliittymän. Sivutilassa oletuksena olevat verkkosivut tuntuvat toimivan nopeasti. Nähtäväksi jää, miten tietokantaa käytettävä verkkosivusto toimii.

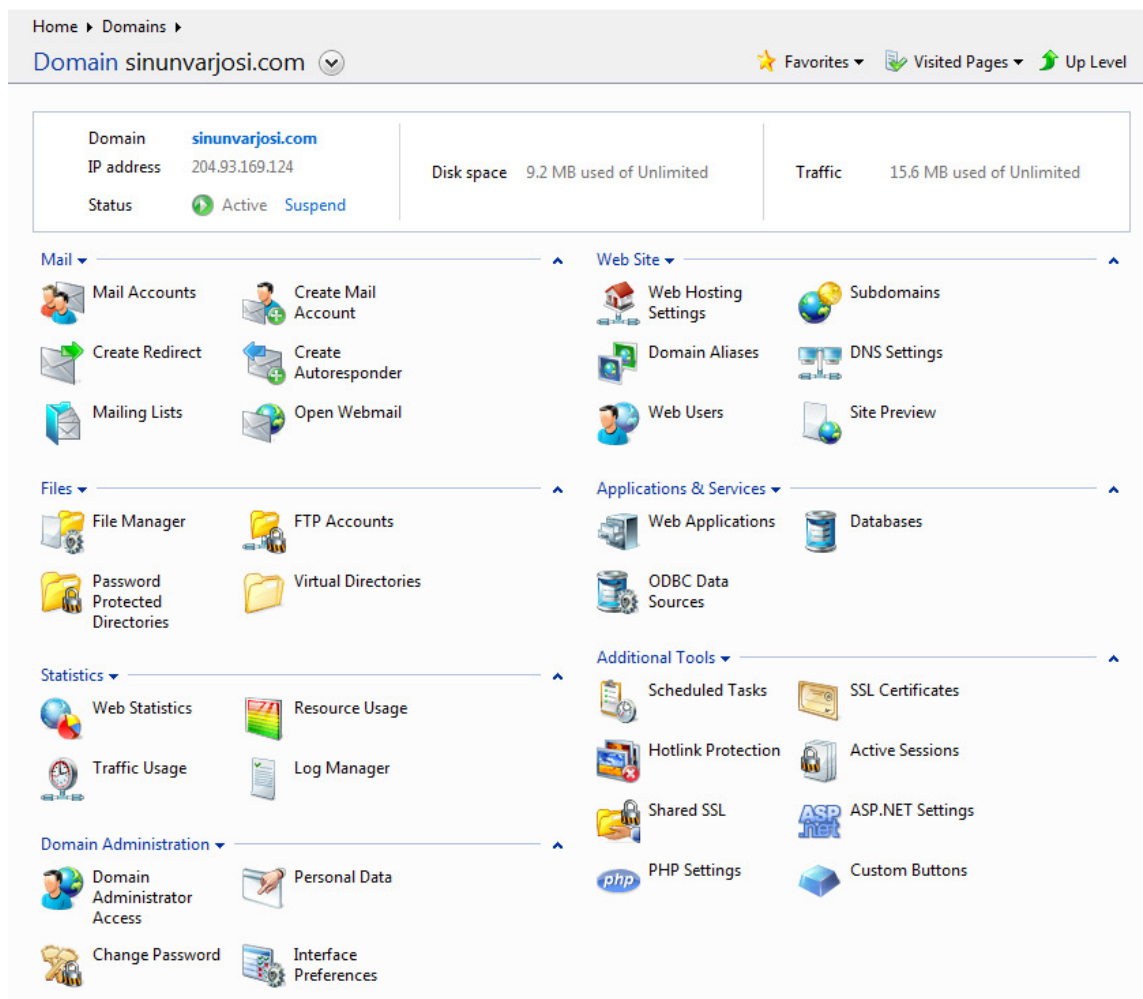
Ongelmat

Julkaisuvaiheessa ongelmia tuotti palveluntarjoajan käyttämä tietokantajärjestelmä, MS SQL Server 2008. Sivustoa kehitettäessä käytettiin saman järjestelmän versiota 2005. Versioristiriidan vuoksi tietokantaa ei voitu kopioida palveluntarjoajan palvelimelle. Julkaisua varten jouduttiin päivittämään kehitystyökaluja. Päivittäminen ei ollut täysin ongelmaton, koska tietokantahallintajärjestelmän myötä jouduttiin päivittämään myös Visual Studio. Muutaman tunnin päivityksen jälkeen ohjelmistot olivat ajan tasalla.

Vaiheet

Maksun jälkeen palveluntarjoajalta saatiin sähköpostiviesti, josta löytyi linkki sivutilan hallintatyökaluun, käyttäjätunnukset, nimipalvelimien osoitteet ja muut tiedot, mitä sivujen julkaisuun vaaditaan. Hallintapaneelissa ensimmäinen toimenpide oli luoda sivusto nimeltä sinunvarjosi.com, jotta samalta palveluntarjo-

ajalta hankittava samanniminen verkkotunnus ohjautuisi oikeaan paikkaan. Sivuston luonti onnistui ongelmitta. Kyseinen palveluntarjoaja mahdollistaa usean verkkosivuston luonnin yhden käyttäjätilin alle. Sivuston luonnin jälkeen hallintatyökalussa päästään näkemään kaikki sivustoon liittyvät tiedot ja ominaisuudet (kuva 14).



Kuva 14 Palveluntarjoajan hallintatyökalu

Jokaisesta erillisestä sivustosta nähdään muun muassa levytilan käyttö, kuukausittainen liikenne, sähköpostitilit, sivustokohtaiset asetukset, tiedostot, tilastot ja tietokannat. Tietokannat kopioitiin palveluntarjoajan palvelimelle Microsoft SQL Server Management Studio -ohjelmalla (ks. luku 2.6). Ohjelmalla otettiin ensin varmuuskopio paikallisesta tietokannasta. Tämän jälkeen otettiin etäyhteys etäpalvelimeen, jonne voitiin palauttaa varmuuskopioitu tietokanta. Sivuston tiedostot olisi voitu siirtää palvelimelle hallintatyökalun avulla, mutta

tähän tarkoitukseen käytettiin FTP-tiedonsiirtoa tukevaa ohjelmaa. Tiedostojen siirtäminen FTP:n avulla on huomattavasti käyttäjäystävällisempää, koska ohjelma tallentaa kaikki tiedonsiirtoon käytetyt tunnukset ja asetukset. Sekä tietokannan että tiedostojen ollessa valmiina palvelimella rekisteröitiin verkkotunnus. Tunnuksen rekisteröinnin jälkeen kului lähes kaksi vuorokautta, ennen kuin tunnus rekisteröityi ympäri Internetiä ja sivusto saatiin näkyviin selaimella. Viimeisenä julkaisuvaiheen toimenpiteenä luotiin sähköpostitili verkkotunnukselle. Sähköpostitilin toiminta varmistettiin lisäämällä se Outlook-ohjelmaan ja lähettämällä testiviesti.

4 POHDINTA

Tässä luvussa pohditaan opinnäytetyön tavoitteita ja niiden toteutumista. Tavoitteena oli tehdä dynaaminen ASP.NET-verkkosovellus. Sivustolle haluttiin työkalu, jolla ylläpitäjä voi hallita sivustolla esiintyviä tietoja. Sivuston julkisivun tulisi hakea sisältö tietokannasta, jotta sisältöä voitaisiin muokata vapaasti.

Työn toteutusta hidasti uusien tekniikoiden kuten jQuery, LINQ to SQL ja ASP.NET samanaikainen opiskelu toteutuksen aikana. Uusien asioiden oppiminen sujui kuitenkin ilman suurempia ongelmia, sillä Internet tarjoaa paljon hyviä esimerkkejä ja informaatiota. ASP.NET on Microsoftin tekniikka, jolloin hyviä ja ajantasaisia ohjeistuksia löydettiin yhtiön virallisilta sivuilta. Tekniikoita opetellessa saatiin selville niiden tarjoamat hyödyt ja rajoitukset. Tulevissa verkkosovelluksissa tekniikoiden käyttö on jo paljon nopeampaa ja luontevampaa.

Olisin voinut tutustua jQueryn tarjoamiin Ajaxia tukeviin toimintoihin aiemmin kuin vasta toteutuksen loppuvaiheilla. Ajaxin avulla varsinkin hallintaosion toimintoja olisi saatu dynaamisemmaksi, ilman turhaa koko sivun uudelleenlataamista tietyissä tilanteissa. Työstä saatiin hyviä ideoita Ajaxin käyttöön tulevia verkkosovelluksia varten. Sivustoa saatetaan vielä myöhemmin kehittää käyttäjäystävällisemmäksi toteuttamalla Ajax-toimintoja.

Sivuston ohjelmakoodissa löytyisi varmasti optimoinnin varaa. Sivuston toteutustekniikalla ja käytetyllä arkkitehtuurilla sama asia voidaan tehdä ohjelmakoodissa monella eri tavalla. Jotkin asiat ohjelmakoodissa jäivät toteutetuksi tavalla, jolla ne saatiin ensimmäisellä yrityksellä toimimaan, vaikka toiminnon tekoon löytyisi tehokkaampikin tapa. Tämä johtuu siitä, että tekniikkaa opeteltiin koko ajan samalla, eikä ollut tiettyä mallia, kuinka kukin toiminto kannattaa toteuttaa.

Lopputuloksena saatiin kaikin puolin toimiva verkkosivusto. Julkisivun ulkoasu toimii eri selaimilla moitteettomasti. Hallintaosio on toimiva ja täyttää vaatimukset, jotka sille asetettiin. Hallintaosiosta voidaan hallita selkeästi sivuston sisältöä. Sivuston selaaminen toimii nopeasti, vaikka sivutila ei olekaan hankittu Suomesta ja jokainen sivu hakee sisältönsä tietokannasta.

Itse olen tyytyväinen lopputulokseen. Mielestäni on hyödyllistä opiskella itsenäisesti uusia tekniikoita, eikä tehdä asioita aina tutulla tavalla. Aloin kehittämään verkkosivustoa aluksi tutulla PHP-tekniikalla, kunnes muutaman työtunnin jälkeen huomasin, etten tulisi mahdollisesti oppimaan mitään uutta. Jatkossa aion painottua tekemään verkkosivustoja PHP:tä enemmän ASP.NET-tekniikalla ja MVC-arkkitehtuurilla. Työn toteutuksen loppupuolella arkkitehtuurista julkaistiin uusi versio, MVC 2, joka tarjoaa uusia ominaisuuksia. Arkkitehtuurin uusi versio onkin seuraava itseopiskelun aihe.

KUVAT

- Kuva 1. MVC-arkkitehtuurin osat, s. 9
- Kuva 2. Esimerkin taulu, s. 18
- Kuva 3. Tietokantarakenne, s. 21
- Kuva 4. iColorPicker-komponentti, s. 22
- Kuva 5. markItUp!-komponentti, s. 23
- Kuva 6. Sisällön hallinta, s. 24
- Kuva 7. Elementin muokkaus, s. 25
- Kuva 8. Tuotteiden hallinta, s. 27
- Kuva 9. Ryhmän muokkaus, s. 28
- Kuva 10. Mallin muokkaus, s. 29
- Kuva 11. Palautteiden hallinta, s. 31
- Kuva 12. Julkisivu, s. 33
- Kuva 13. Varjostinmallien listaus julkisivulla, s. 34
- Kuva 14. Palveluntarjoajan hallintatyökalu, s. 36

TAULUKOT

- Taulukko 1. Tietokannan taulun rakenne, s. 15

LÄHTEET

CSS Wikipedia. Cascading Style Sheets. Luettu 15.3.2010.
http://fi.wikipedia.org/wiki/Cascading_Style_Sheets

Esposito D. 2008. Programming Microsoft ASP.NET 3.5. Washington: Microsoft Press.

Gertz M. 2000. Oracle/SQL Tutorial. Luettu 22.3.2010.
<http://dbis.ucdavis.edu/courses/sqltutorial/tutorial2.pdf>

jQuery Wikipedia. jQuery. Luettu 16.4.2010. <http://en.wikipedia.org/wiki/JQuery>

Järvinen J. 2008. Linq helpottaa sql-kyselyjä. Luettu 28.3.2010.
http://www.tietokone.fi/lehti/tietokone_9_2008/linq_helpottaa_sql_kyselyja_730

Korpela J. 2004. XHTML-käsikirja. Jyväskylä: Docendo Finland Oy.

Laine H. 2000. SQL-kyselyt. Luettu 28.3.2010.
<http://www.cs.helsinki.fi/u/laine/tikape/moniste3.pdf>

LINQ to SQL MSDN. LINQ to SQL: .NET Language-Integrated Query for Relational Data. Luettu 28.3.2010.
<http://msdn.microsoft.com/en-us/library/bb425822.aspx>

MS ASP.NET. What is ASP.NET MVC? Luettu 20.4.2010.
<http://www.asp.net/mvc/whatisaspmvc>

MVC Wikipedia. MVC-arkkitehtuuri. Luettu 20.4.2010.
<http://fi.wikipedia.org/wiki/MVC-arkkitehtuuri>

Myhosting. Myhosting.com wiki. Microsoft SQL. Luettu 15.1.2010.
http://myhosting.com/wiki/Microsoft_SQL

Salvat J. 2010. markItUp! Universal markup editor. Luettu 18.4.2010.
<http://markitup.jaysalvat.com>

SQL 2Kmediat. SQL-opas. Luettu 22.3.2010.
<http://www.2kmediat.com/sql/alkeet.asp>

SQL Server Express Readme Microsoft. Luettu 22.3.2010.
<http://support.microsoft.com/kb/910229>

SSMSE Microsoft. Luettu 22.3.2010.
<http://www.microsoft.com/downloads/details.aspx?FamilyId=C243A5AE-4BD1-4E3D-94B8-5A0F62BF7796&displaylang=en#filelist>

SuperSite. jQuery-Color-Picker. Luettu 18.4.2010.
http://www.supersite.me/index.php?p=2_1

XML Wikipedia. XML. Luettu 12.3.2010. <http://fi.wikipedia.org/wiki/XML>